이변량 자료 탐색

이변량 자료 탐색

- 이변량 자료의 분석
  - 각 변수의 개별 분포 파악
  - 두 변수의 분포 비교
  - 두 변수의 관계 탐색

- 이변량 범주형 자료
  - 막대 그래프: 쌓아 올린 형태, 옆으로 붙여 놓은 형태
  - Mosaic plot

- 이변량 연속형 자료
  - 분포 비교를 위한 그래프
  - 관계 탐색을 위한 그래프

1. 연속형 변수의 분포를 비교하기 위한 그래프

- 예제: mpg의 변수 cyl에 따른 hwy의 분포 비교
  - cyl로 구분되는 그룹에 속한 자료의 개수

```
> mpg %>% group_by(cyl) %>% summarize(n=n())
# A tibble: 4 x 2
    cyl     n
  <int> <int>
1     4    81
2     5     4
3     6    79
4     8    70
```

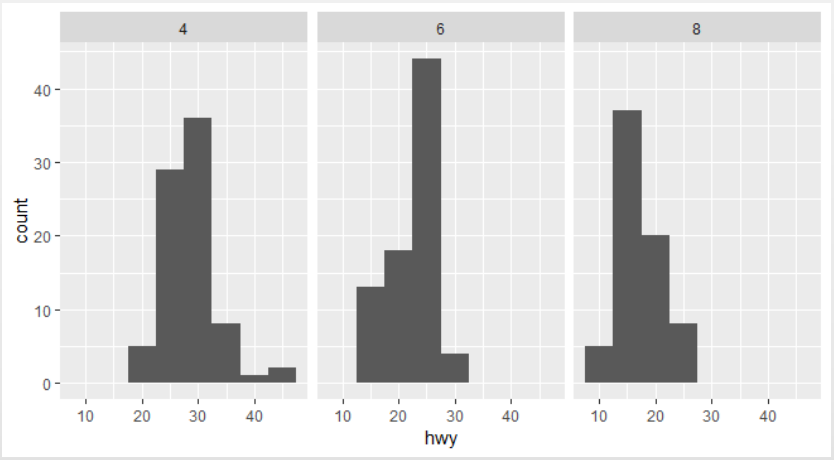
  - cyl이 5가 되는 자료의 개수가 너무 작음
  - cyl이 4, 6, 8인 그룹에 대해서만 hwy의 분포 비교

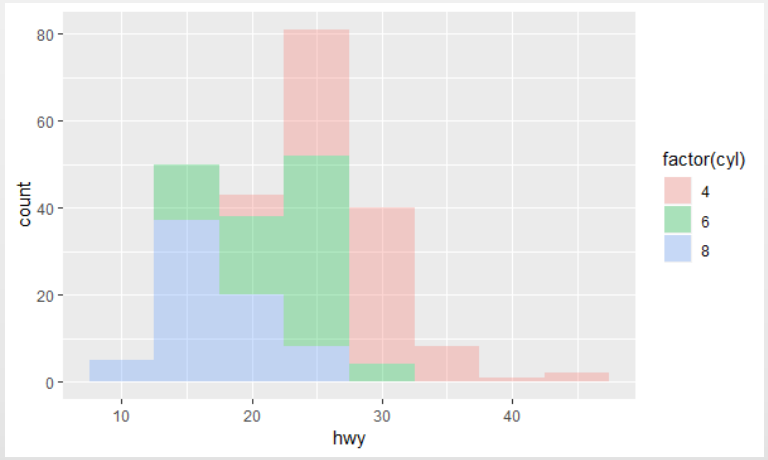
```
> mpg_1 <- mpg %>% filter(cyl!=5)
```

- 히스토그램에 의한 그룹 자료의 분포 비교

```
> ggplot(mpg_1, aes(x=hwy)) +
    geom_histogram(binwidth=5) +
    facet_wrap(~ cyl)
```

```
> ggplot(mpg_1, aes(x=hwy, fill=factor(cyl))) +
+   geom_histogram(binwidth=5, alpha=0.3)
```



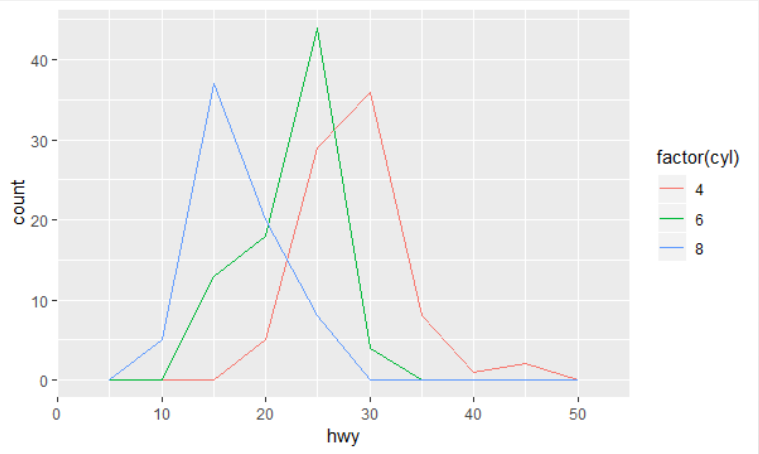


그룹간 분포 비교가 쉽지 않은 그래프

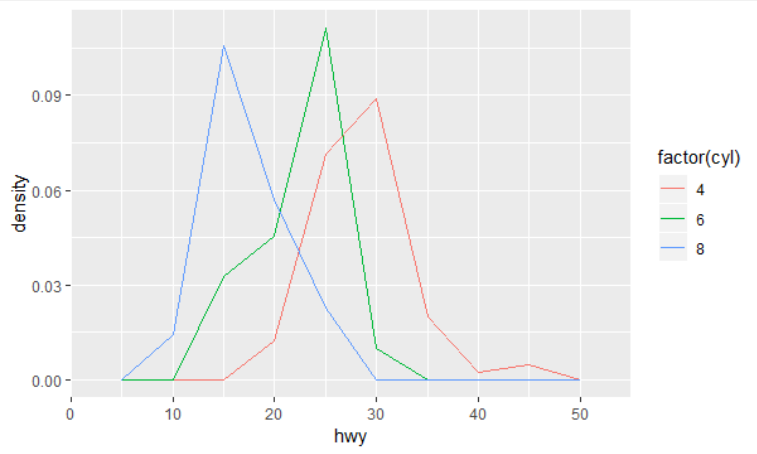
- 도수분포다각형에 의한 그룹 자료의 분포 비교

```
> p <- ggplot(mpg_1, aes(x=hwy, color=factor(cyl)))
> p + geom_freqpoly(binwidth=5)
> p + geom_freqpoly(aes(y=..density..), binwidth=5)
```

도수분포다각형



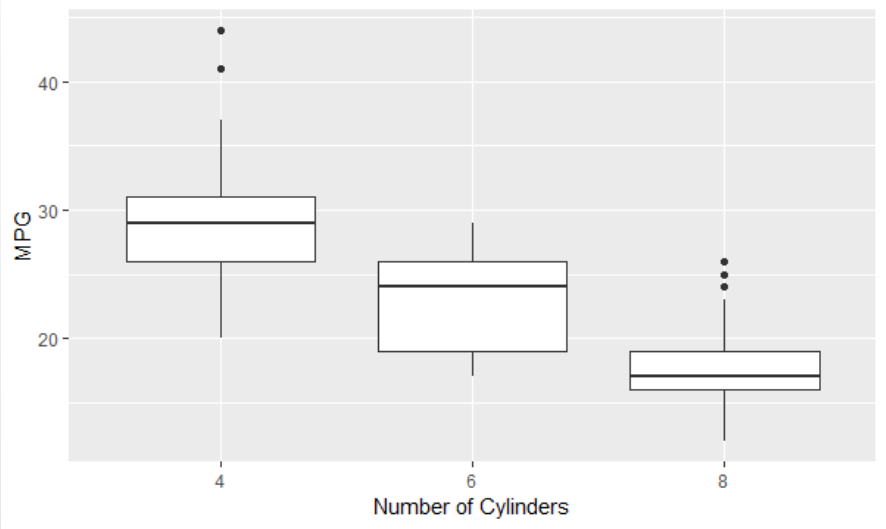
상대도수분포다각형



- 각 그룹에 속한 자료의 개수가 다름
- 상대도수를 이용하는 것이 더 효과적인 비교가 가능

- 상자그림에 의한 그룹 자료의 분포 비교

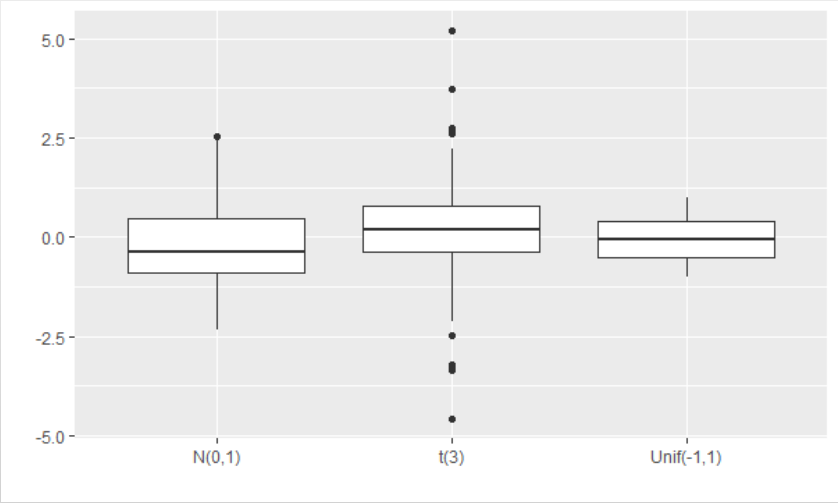
```
> ggplot(mpg_1, aes(x=factor(cyl), y=hwy)) +
    geom_boxplot() +
    labs(x="Number of Cylinders", y="MPG")
```



- 상자그림에 의한 그룹 자료의 분포 비교

  - 표준정규분포, t(3) 분포, Unif(-1,1)에서 각각 100개 난수 추출
  - 세 자료의 분포를 상자그림으로 비교

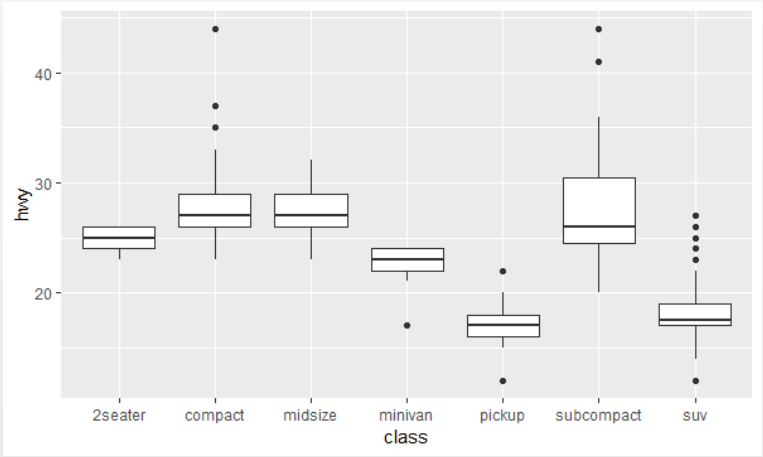
```
> set.seed(1234)
> x1 <- rnorm(100)
> x2 <- rt(100, df=3)
> x3 <- runif(100, min=-1, max=1)
> data.frame(x=c(rep(1:3,each=100)), y=c(x1,x2,x3)) %>%
    ggplot(aes(factor(x), y)) +
    geom_boxplot() +
    scale_x_discrete(labels=c("N(0,1)","t(3)","Unif(-1,1)")) +
    labs(x="", y="")
```



- 상자그림에 의한 그룹 자료의 분포 비교

  - mpg의 변수 hwy의 상자그림을 class의 수준별로 작성

```
> ggplot(mpg, aes(x=class, y=hwy)) +
    geom_boxplot()
```

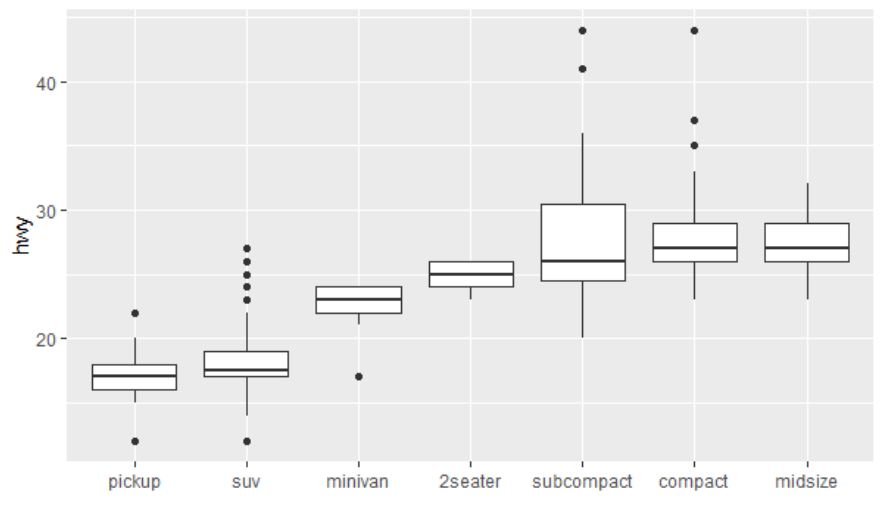


- class의 범주 순서에 따라 상자그림 배열
- hwy의 중앙값에 따라 배열하는 것이 분포 비교에 더 좋음
- class의 범주 수준을 hwy의 중앙값을 기준으로 다시 배열

reorder(class, hwy, FUN=median)

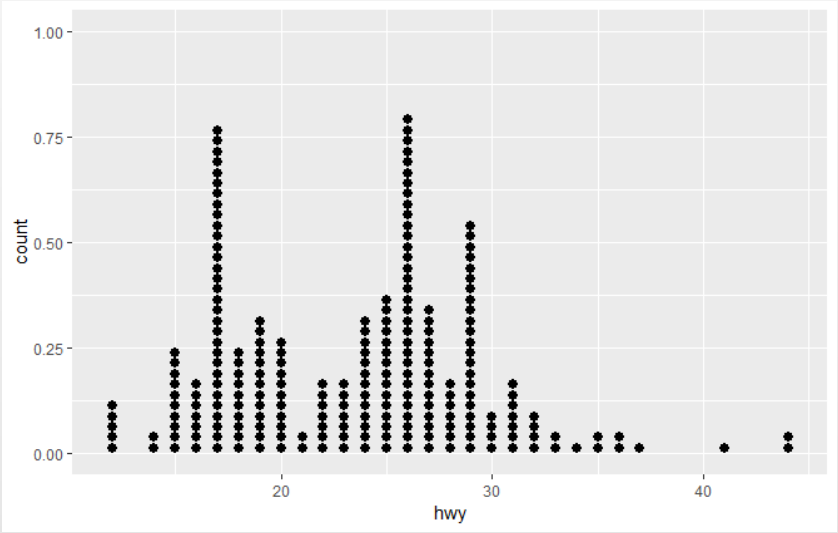
- 상자그림의 배치 순서 조정

```
> ggplot(mpg, aes(x=reorder(class, hwy, FUN=median), y=hwy)) +
    geom_boxplot() +
    labs(x="")
```



- 다중 점 그래프에 의한 그룹 자료의 분포 비교
  - 점 그래프: 변수 hwy

```
> ggplot(mpg, aes(x=hwy)) +
+   geom_dotplot(binwidth=0.5)
```

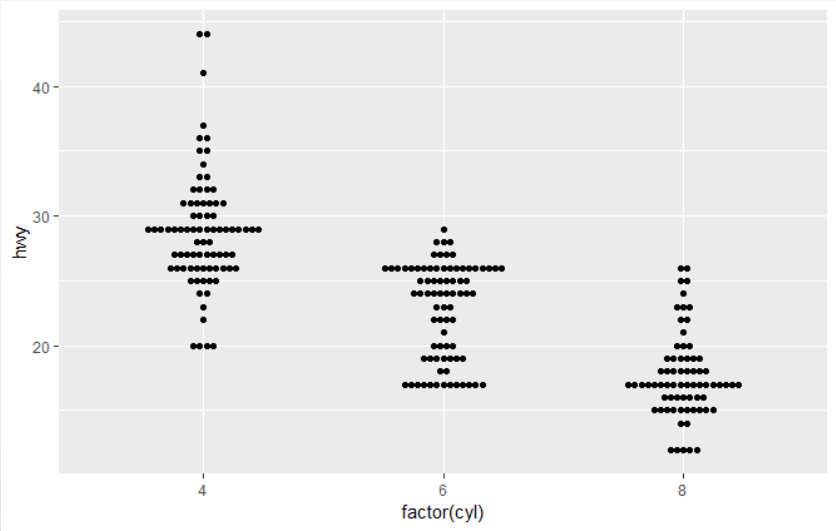


- 자료의 전 범위를 구간으로 구분
- 각 구간에 속한 자료 한 개당 하나의 점을 위로 쌓아 올리는 그래프
- 소규모 자료에 적합

- mpg의 변수 cyl에 따른 hwy의 다중 점 그래프

```
> mpg_1 <- mpg %>% filter(cyl!=5)

> ggplot(mpg_1, aes(x=factor(cyl), y=hwy)) +
    geom_dotplot(binaxis="y", binwidth=0.5, stackdir="center")
```



- binaxis: 구간 설정 대상이 되는 축
  디폴트는 "x"

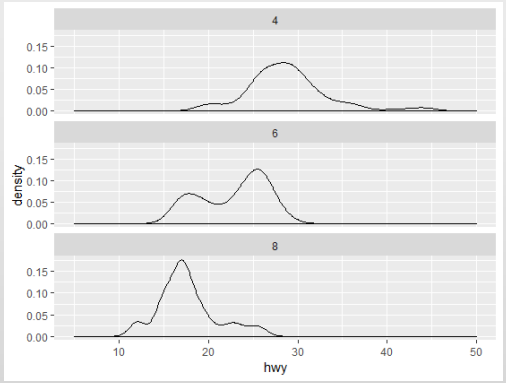
- stackdir: 점을 쌓아 가는 방향
  "up", "down", "center"

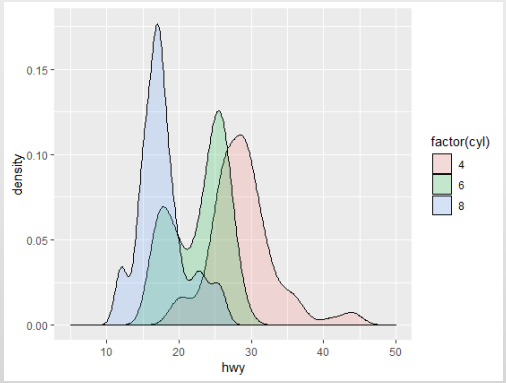
- 확률밀도함수 그래프에 의한 그룹 자료의 분포 비교

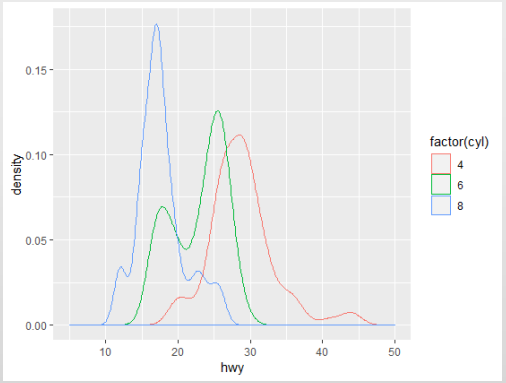
```
> ggplot(mpg_1, aes(x=hwy)) +
    geom_density() +
    xlim(5,50) +
    facet_wrap(~cyl, ncol=1)
```

```
> ggplot(mpg_1, aes(x=hwy, fill=factor(cyl))) +
    geom_density(alpha=0.2) +
    xlim(5,50)
```

```
> ggplot(mpg_1, aes(x=hwy, color=factor(cyl))) +
    geom_density() +
    xlim(5,50)
```



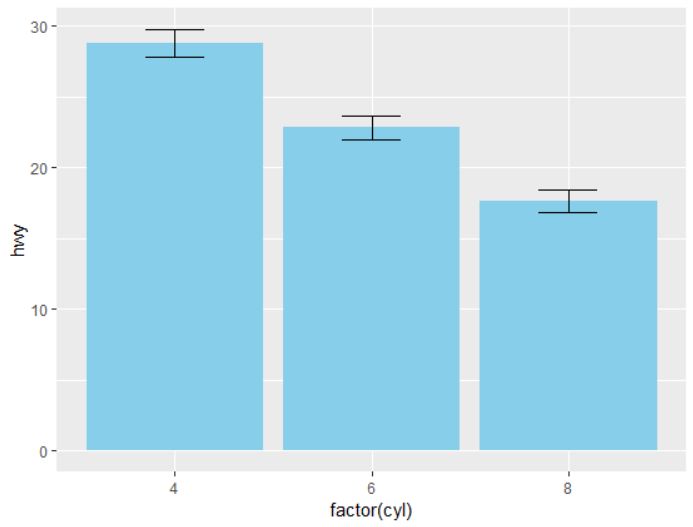




- 평균 막대 그래프와 error bar에 의한 그룹 자료의 평균값 비교

  - 그룹별 자료의 평균 비교에 막대 그래프 이용
  - Error bar: 분포의 변동 혹은 신뢰구간을 표시하는 그래프

    - mpg의 변수 cyl에 따른 hwy의 평균 및 신뢰구간



- 막대 그레프: 변수 cly에 따른 hwy의 평균
- Error bar: 각 그룹별 hwy의 95% 신뢰구간

- 작성 방법

1) 그룹 자료의 평균과 신뢰구간이 주어진 경우

```
> hwy_stat
# A tibble: 3 x 6
    cyl mean_hwy sd_hwy n_hwy ci_low ci_up
  <int>    <dbl>  <dbl> <int>  <dbl> <dbl>
1     4     28.8   4.52    81   27.8  29.8
2     6     22.8   3.69    79   22.0  23.6
3     8     17.6   3.26    70   16.9  18.4
```

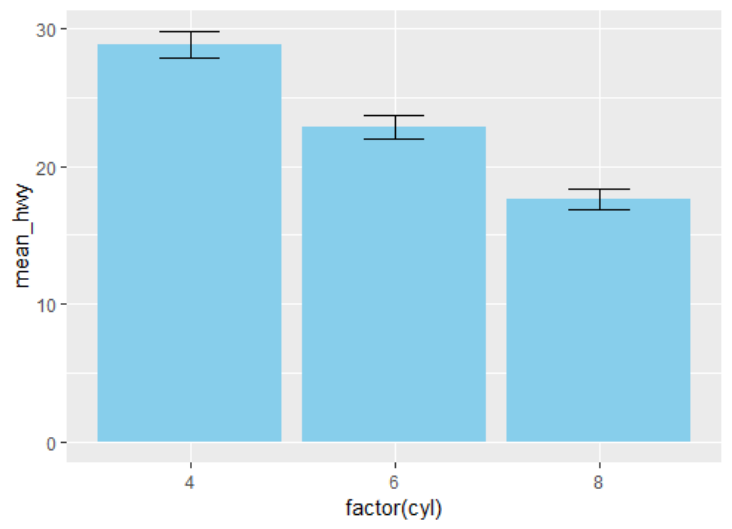
- 평균: mean_hwy
- 신뢰구간 상한: ci_up
- 신뢰구간 하한: ci_low

- hwy_stat의 계산

```
> hwy_stat <- mpg %>%
    filter(cyl!=5) %>%
    group_by(cyl) %>%
    summarize(mean_hwy=mean(hwy), sd_hwy=sd(hwy),
              n_hwy=n(),
              ci_low=mean_hwy-qt(0.975,df=n_hwy-1)*sd_hwy/sqrt(n_hwy),
              ci_up=mean_hwy+qt(0.975,df=n_hwy-1)*sd_hwy/sqrt(n_hwy))
```

- 주어진 요약 통계량 자료로 막대 그래프 및 error bar 작성

```
> ggplot(hwy_stat, aes(x=factor(cyl), y=mean_hwy)) +
    geom_col(fill="skyblue") +
    geom_errorbar(aes(ymin=ci_low, ymax=ci_up), width=0.3)
```



2) 원 자료만 주어진 경우

```
> mpg %>% filter(cyl!=5) %>%
    ggplot(aes(x=factor(cyl), y=hwy)) +
    stat_summary(fun.y="mean", geom="bar", fill="skyblue") +
    stat_summary(fun.data="mean_cl_normal", geom="errorbar",
                 width=0.3)
```

- fun.data: ymin과 y, ymax를 계산할 수 있는 함수 지정

- mean_cl_normal( ): 정규분포 가정에서 모평균의 신뢰구간 계산.
  패키지 Hmisc의 함수 smean.cl.normal( )을 불러 작업

```
> Hmisc::smean.cl.normal(mpg$hwy)
    Mean    Lower    Upper
23.44017 22.67324 24.20710
```

- 연습문제

  - 13장 연습문제 3번
  - 13장 연습문제 4번

2. 연속형 변수의 관계 탐색을 위한 그래프: 산점도

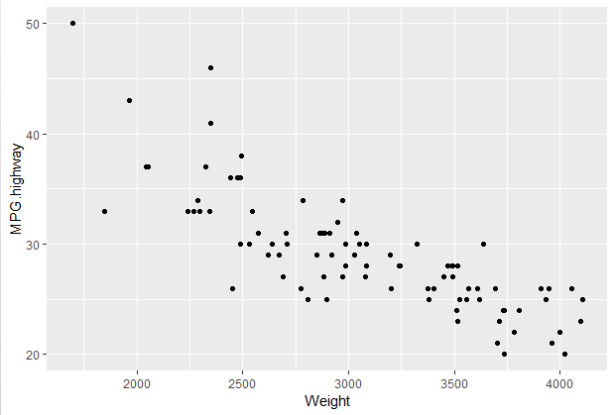
1) 다양한 유형의 산점도 작성

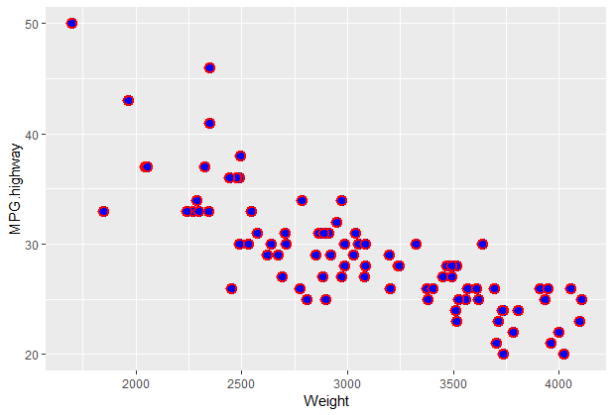
- 기본적인 형태의 산점도: Cars93의 Weight와 MPG.highway

```
> data(Cars93, package="MASS")
```

```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway)) +
+   geom_point()
```

```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway)) +
+   geom_point(shape=21, color="red", fill="blue",
+              stroke=1.5, size=3)
```



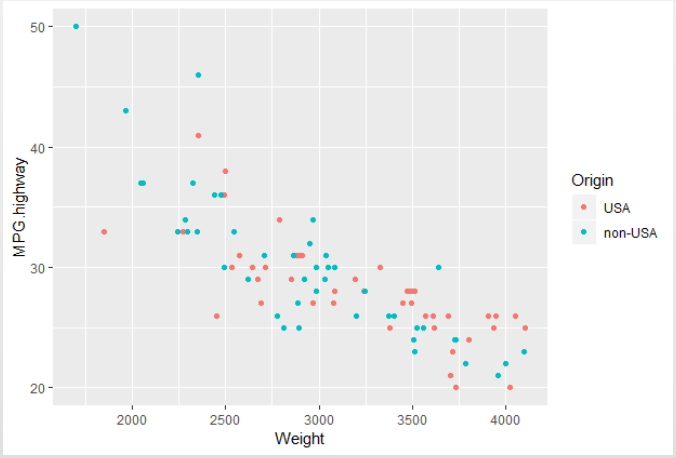


- 시각적 요소에 세 번째 변수 매핑: 산점도에서 세 변수의 관계 탐색

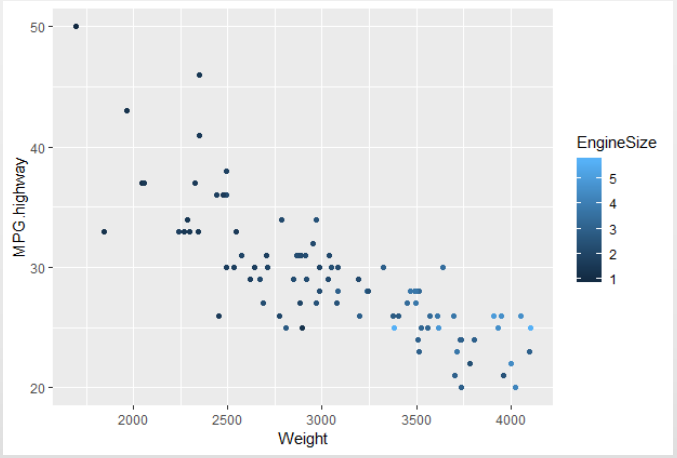
```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway, color=Origin)) +
    geom_point()

> ggplot(Cars93, aes(x=Weight, y=MPG.highway, color=EngineSize)) +
    geom_point()
```

- color에 요인 매핑



- color에 숫자형 변수 매핑

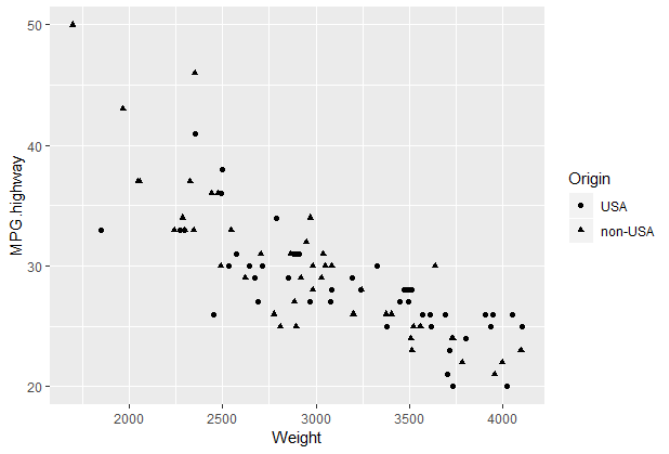


시각적 요소 color에 숫자형 변수의 매핑은 좋은 선택이 아닌 것으로 보임

- shape에 요인 및 숫자형 변수 매핑

```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway, shape=Origin)) +
    geom_point()
```

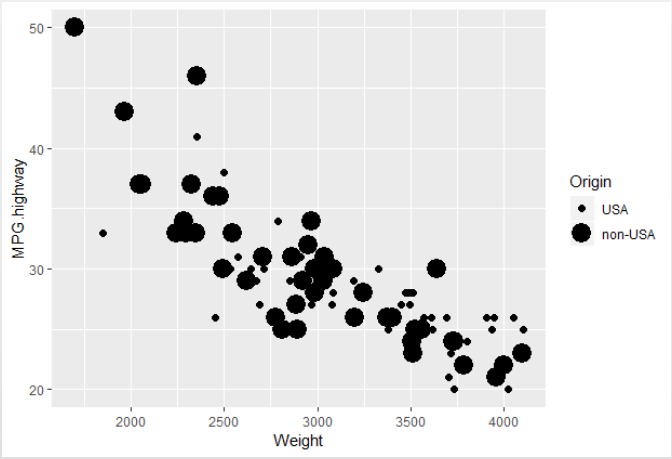
```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway, shape=EngineSize)) +
    geom_point()
Error: A continuous variable can not be mapped to shape
```

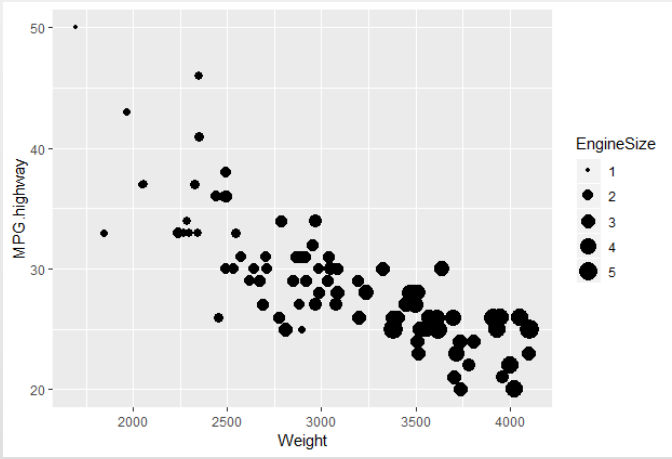


- size에 요인 및 숫자형 변수 매핑

```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway, size=Origin)) +
    geom_point()
Warning message:
Using size for a discrete variable is not advised.
```

```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway, size=EngineSize)) +
    geom_point()
```

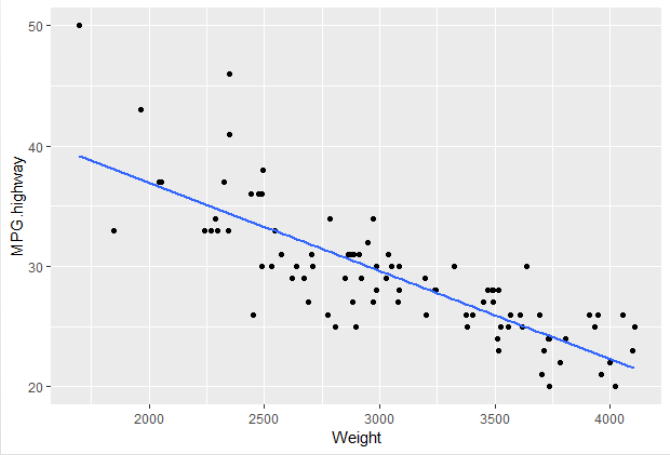


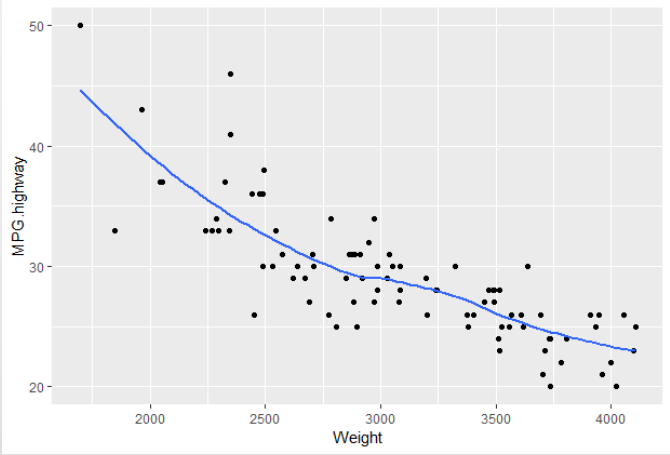


- 산점도에 회귀직선 추가

```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway)) +
    geom_point() +
    geom_smooth(method="lm", se=FALSE)

> ggplot(Cars93, aes(x=Weight, y=MPG.highway)) +
    geom_point() +
    geom_smooth(se=FALSE)
```





- 회귀직선과 비모수 회귀곡선을 함께 산점도에 추가

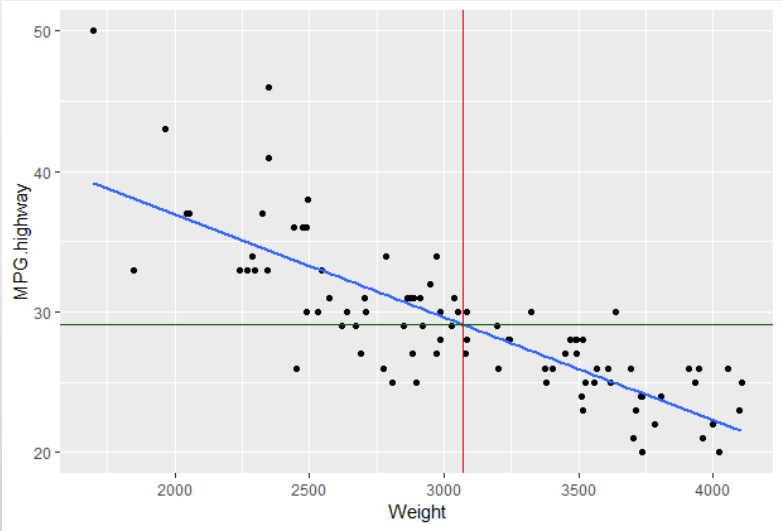
```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway)) +
    geom_point() +
    geom_smooth(aes(color="lm"), method="lm", se=FALSE) +
    geom_smooth(aes(color="loess"), se=FALSE) +
    labs(color="Method")
```

- 시각적 요소에 문자열 매핑
- 두 종류의 선에 legend를 추가하는 유용한 방법

- 산점도에 수평선, 수직선 추가

  - 직선 추가 함수: geom_abline(slope, intercept)
  - 수직선 추가 함수: geom_vline(xintercept)
  - 수평선 추가 함수: geom_hline(yintercept)

```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway)) +
   geom_point() +
   geom_smooth(method="lm", se=FALSE) +
   geom_vline(aes(xintercept=mean(Weight)), color="red") +
   geom_hline(aes(yintercept=mean(MPG.highway)), color="darkgreen" )
```

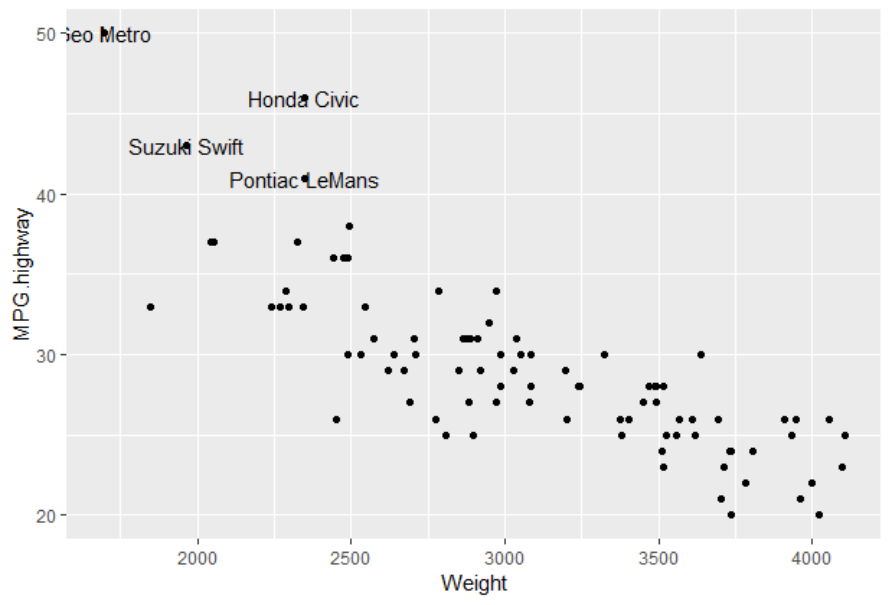


- 산점도
- 회귀직선
- 변수 Weight의 평균에 수직선
- 변수 MPG.highway의 평균에 수평선

- 산점도의 점에 라벨 추가

  - Weight와 MPG.highway의 산점도
  - MPG.highway > 40 인 점에 라벨 추가
  - 라벨 내용: Manufacturer와 Model의 값 결합한 것

```
> p <- ggplot(Cars93, aes(x=Weight, y=MPG.highway)) +
    geom_point()
> p + geom_text(data=filter(Cars93, MPG.highway>40),
                aes(label=paste(Manufacturer, Model)))
```

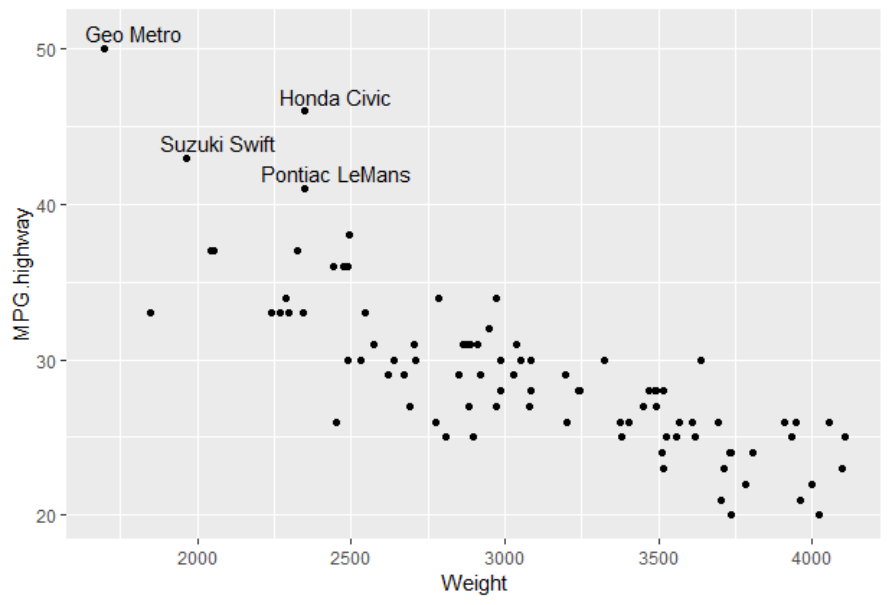


- 라벨의 위치 조정이 필요

- 라벨 위치 조정
  ① vjust, hjust
  ② nudge_x, nudge_y

- 라벨 위치 조정: nudge_x & nudge_y 이용

```
> p + geom_text(data=filter(Cars93, MPG.highway>40),
                aes(label=paste(Manufacturer, Model)),
                nudge_y=1, nudge_x=100)
```



- nudge_x:
  양의 값: 우측으로 이동
  음의 값: 좌측으로 이동

- nudge_y:
  양의 값: 위로 이동
  음의 값: 아래로 이동

- 산점도에 주석 추가
  - Weight와 MPG.highway의 산점도
  - 회귀직선 추가
  - 결정계수 주석으로 추가

결정계수 계산

```
> fit <- lm(MPG.highway~Weight, Cars93)
> r2 <- round(summary(fit)$r.squared, 2)
```

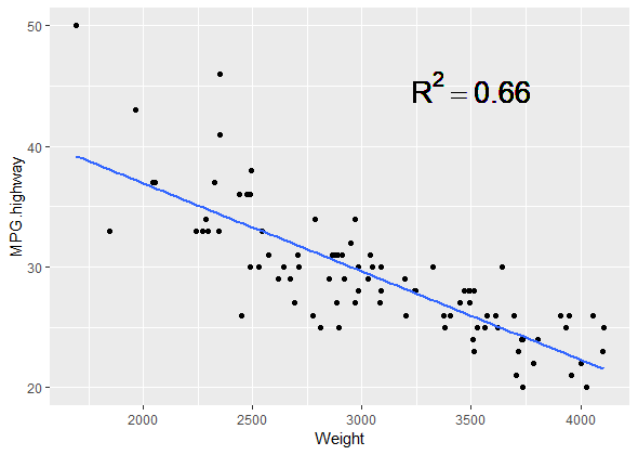
산점도 작성 및 회귀직선 추가

```
> pp <- ggplot(Cars93, aes(x=Weight, y=MPG.highway)) +
    geom_point() +
    geom_smooth(method="lm", se=FALSE)
```

- 결정계수 주석으로 추가

```
> pp +  geom_text(x=3500, y=45, size=7,
                  label=paste("R^2=", r2))
```

```
> pp +  geom_text(x=3500, y=45, size=7,
                label=paste("R^2==", r2), parse=TRUE)
```

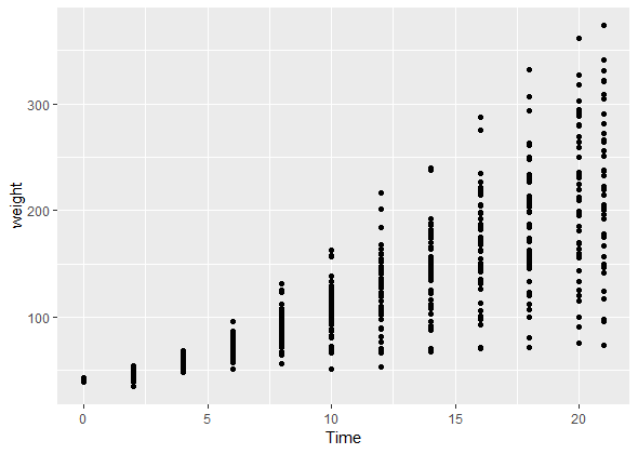


2) 산점도에서 점이 겹쳐지는 문제

- 대규모 자료인 경우
- 두 변수 중 한 변수가 이산형인 경우
- 자료가 반올림된 경우

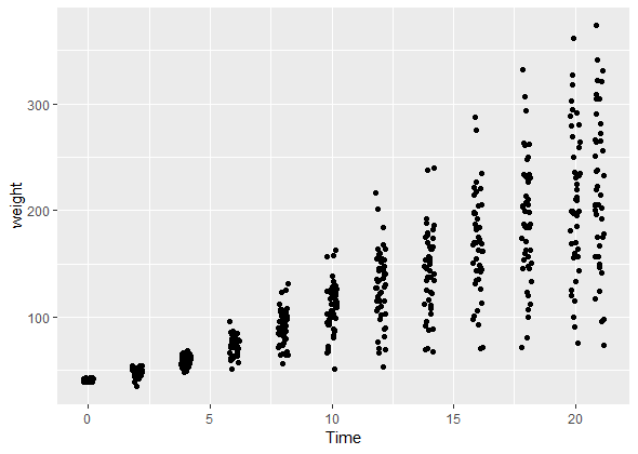
① 한 변수가 이산형인 경우의 예: Chickweight의 변수 Time과 weight

```
> p1 <- ggplot(ChickWeight, aes(x=Time, y=weight))
> p1 + geom_point()
```



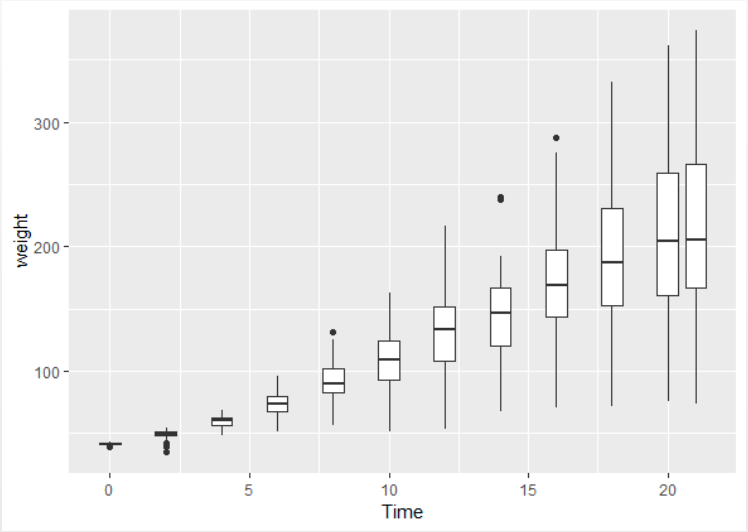
- 대안 1: jittering

```
> p1 + geom_jitter(width=0.2, height=0)
```



- 대안 2: 상자그림

```
> p1 + geom_boxplot(aes(group=Time))
```

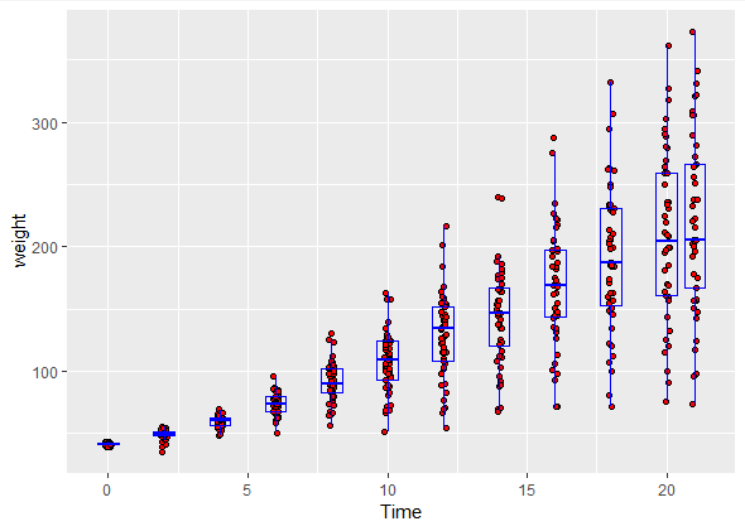



```
> class(ChickWeight$Time)
[1] "numeric"
```

- x 변수인 Time이 숫자형 변수
- 시각적 요소 group에 x를 매핑

- 대안 3: 상자그림과 jittering

```
> p1 + geom_jitter(width=0.1, fill="red", shape=21) +
    geom_boxplot(aes(group=Time), outlier.shape=NA,
                 fill=NA, color="blue")
```



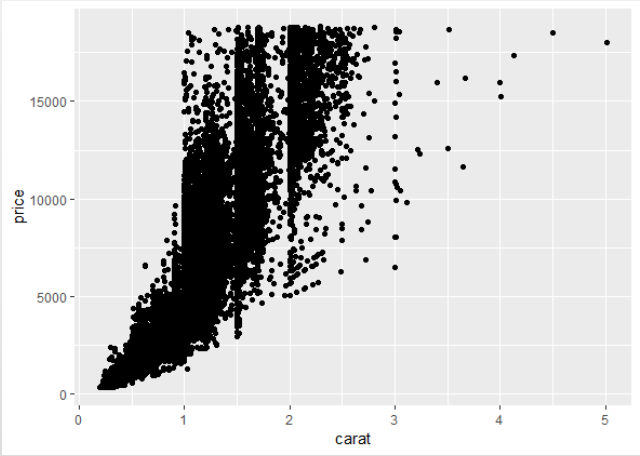
fill=NA
- 상자그림 내부의 흰색 제거
- geom_boxplot을 먼저 실행하고, 그 위에 점을 jittering하면 필요 없음

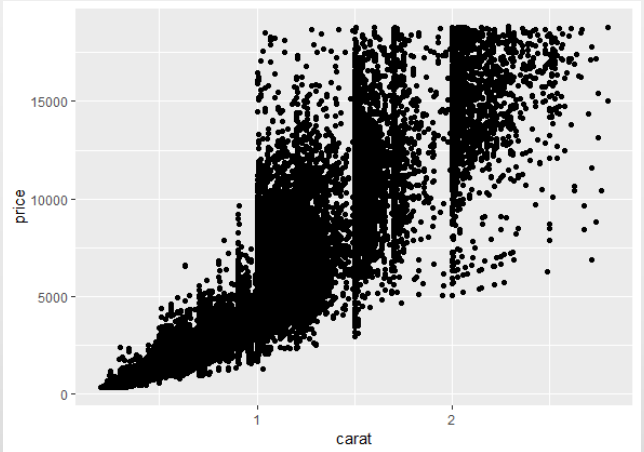
② 대규모 자료의 예: diamonds의 변수 carat과 price의 산점도

```
> ggplot(diamonds, aes(x=carat, y=price)) +
    geom_point()
```

carat<3인 자료만을 대상으로 산점도 작성

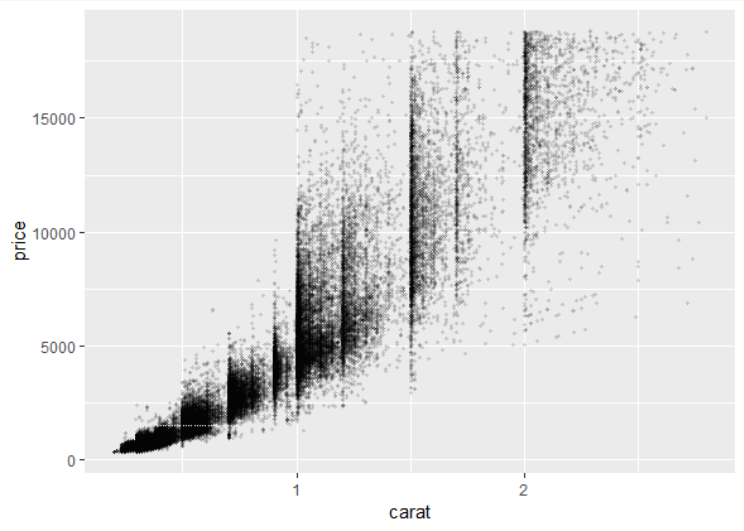
```
> p2 <- ggplot(filter(diamonds, carat<3), aes(x=carat, y=price))
> p2 + geom_point()
```





- 대안 1: 점의 크기를 줄이고 투명도를 높이는 것

```
> p2 + geom_point(alpha=0.1, shape=20)
```

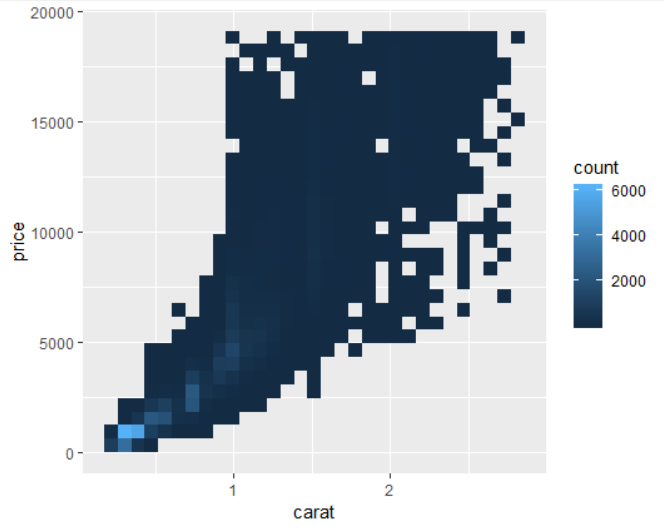


- 여전히 자료의 분포를 정확하게 알아보기 어려운 상황
- 특정 carat의 값에서 점들이 형성하는 수직의 띠 개수가 늘어남

- 대안 2: geom_bin2d( )로 2차원 히스토그램 작성

  - XY 2차원 공간을 직사각형의 영역으로 구분
  - 각 영역에 속한 자료의 개수를 색으로 표현

```
> p2 + geom_bin2d()
```

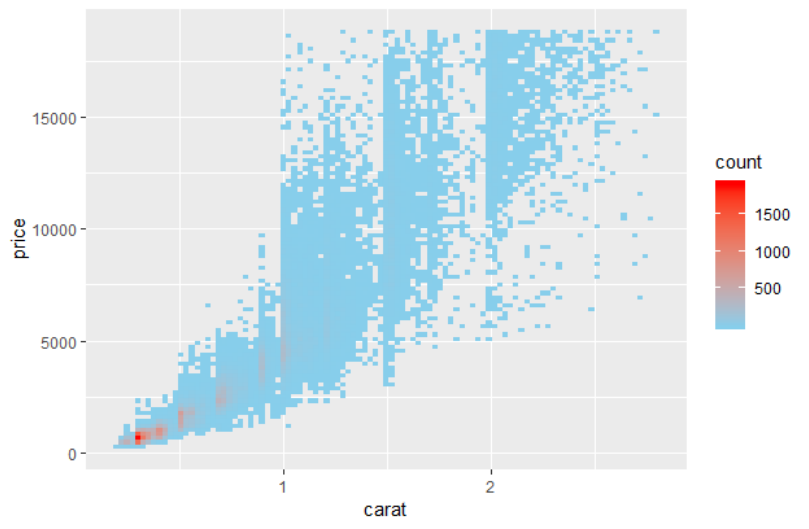


- X축과 Y축을 30개 구간으로 구분
- 전체 공간을 30X30=900의 직사각형으로 구분(디폴트)
- 각 영역에 속한 자료의 개수를 색으로 표현

- 색에 큰 변화가 없어서 구분이 어려움
- 영역을 더 세분화할 필요가 있음

- 구간 개수 늘리고 색에 변화를 준 그래프

```
> p2 + geom_bin2d(bins=100) +
    scale_fill_gradient(low="skyblue", high="red")
```



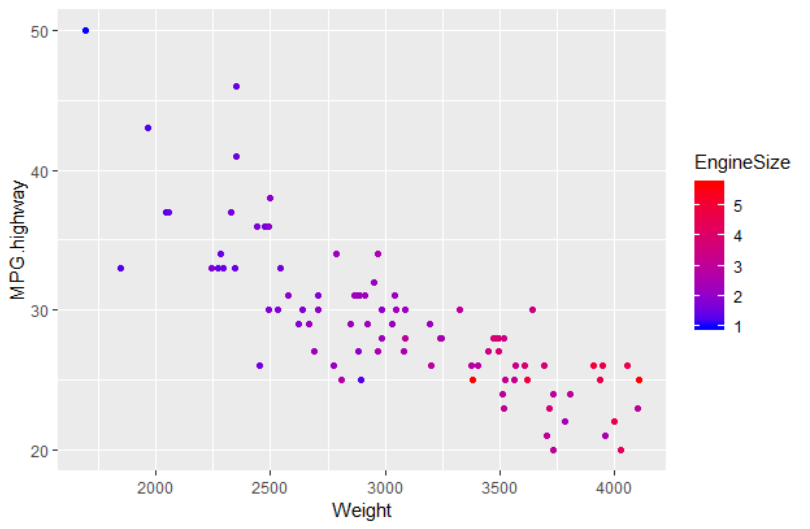
scale_fill_gradient( ):
- two color(low-high) gradient 생성

- color gradient: 각각의 점이 다른 컬러 값을 갖는 일종의 축

- 시작점과 끝점을 지정함으로써 다른 색의 표현이 가능

- scale_*_gradient의 다른 적용 예: Cars93의 Weight vs MPG.highway 산점도

```
>  ggplot(Cars93, aes(x=Weight, y=MPG.highway,
                       color=EngineSize)) +
    geom_point() +
    scale_color_gradient(low="blue", high="red")
```

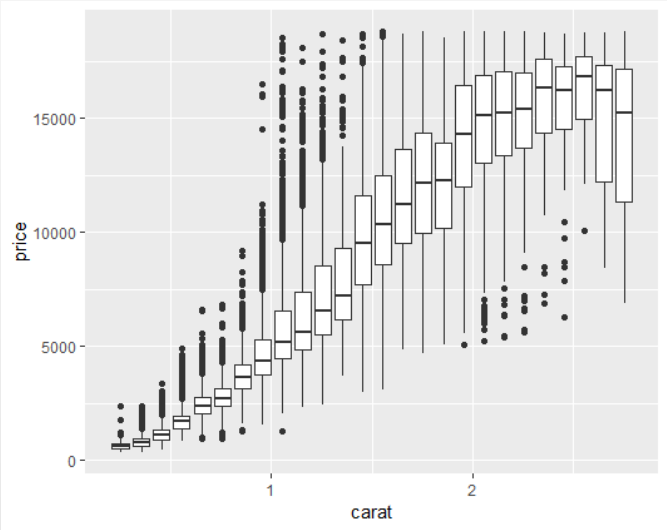


연속형 변수 EngineSize를 color에 매핑

- 대안 3: X축 변수를 범주형으로 변환하고 상자그림 작성
  - 숫자형 변수를 요인으로 전환하는 함수: cut( )을 이용한 함수
    - cut_width(x, width, boundary): 동일 간격으로 구분. 옵션 boundary는 시작점 지정.
    - cut_number(x, number=n): n개 구간으로 구분하되, 각 구간에 속한 자료의 개수를 동일하게 유지
    - cut_interval(x, n, length): n개 구간으로 구분하되, 각 구간의 길이를 length로 동일하게 유지

- 변수 carat을 시작점을 0, 간격을 0.1로 하는 구간으로 구분
- 각 구간의 자료를 대상으로 side-by-side boxplot 작성

```
> p2 + geom_boxplot(aes(group=cut_width(carat,
                                        width=0.1, boundary=0)))
```



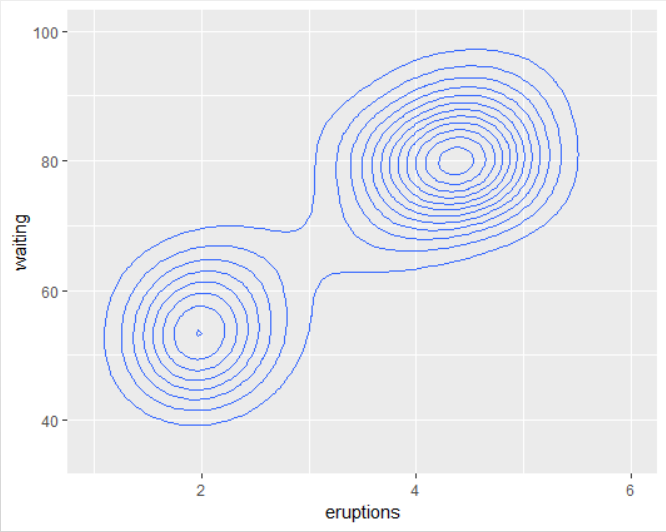
3) 이차원 결합확률밀도 그래프

- 두 연속형 변수의 관계 탐색에서 큰 역할을 할 수 있는 그래프

- 예: faithful의 eruptions와 waiting의 결합확률밀도 추정

```
> p3 <- ggplot(faithful, aes(x=eruptions, y=waiting)) +
    xlim(1,6) + ylim(35,100)
```

```
> p3 + geom_density_2d()
```



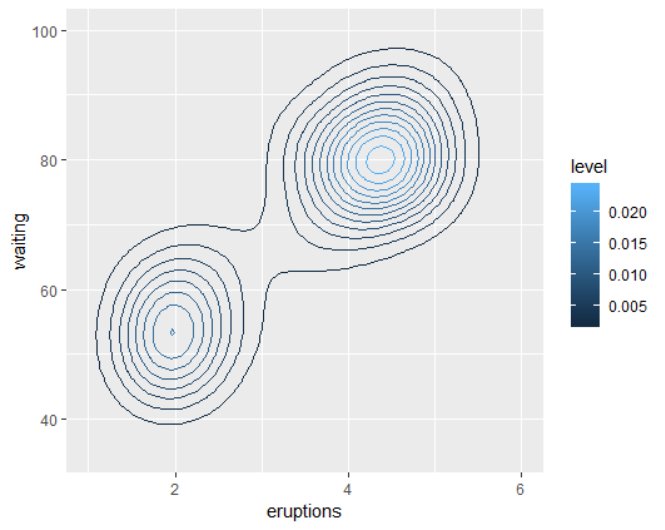
- 등고선 그래프
- 각 등고선에 대한 적절한 라벨 필요
- 색으로 높이를 구분하는 방법

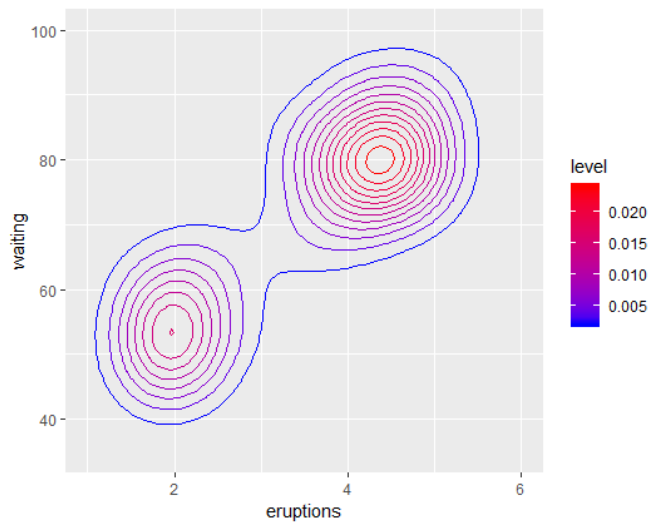
- 색으로 등고선의 높이 표현

```
> p3 + geom_density_2d(aes(color=stat(level)))
```

- stat(level): ..level.. 과 동일
  geom_density_2d( )에서 계산한 변수(등고선 높이)

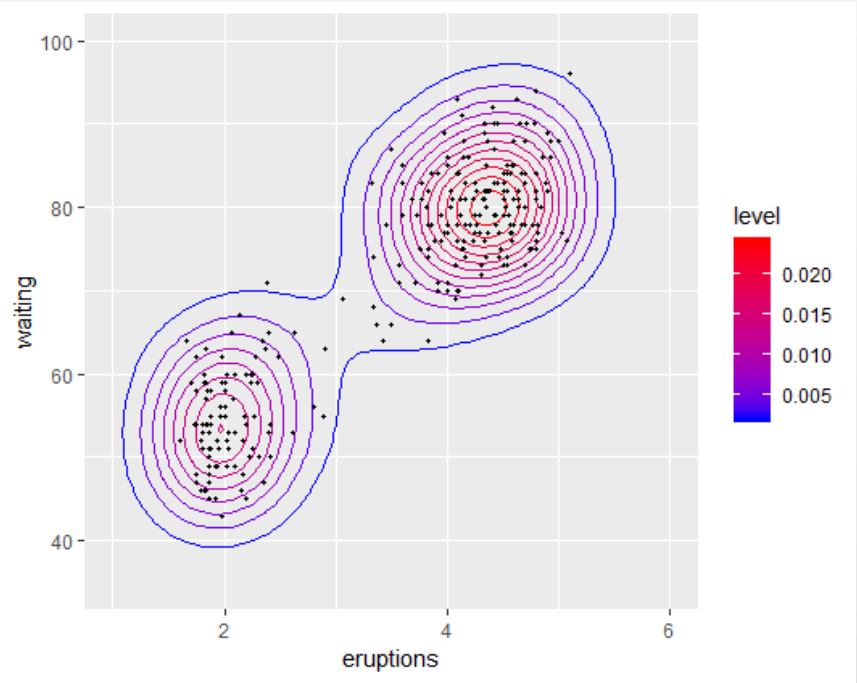
```
> p3 + geom_density_2d(aes(color=stat(level))) +
  scale_color_gradient(low="blue", high="red")
```





- 두 변수의 산점도에 등고선 그래프 추가

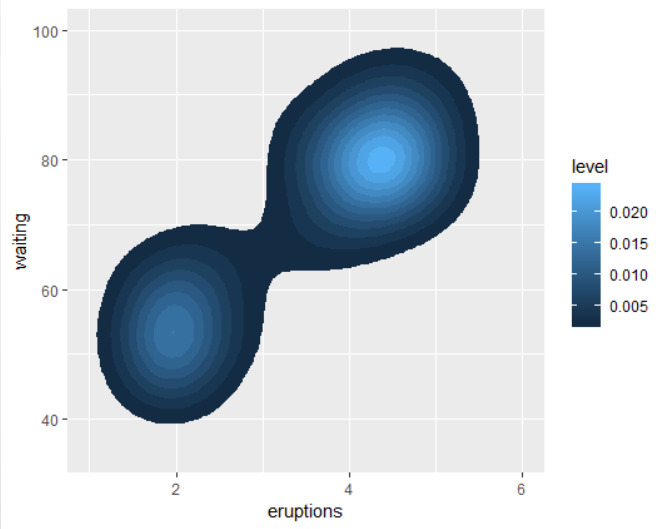
```
> p3 + geom_density_2d(aes(color=stat(level))) +
    scale_color_gradient(low="blue", high="red") +
    geom_point(shape=20)
```

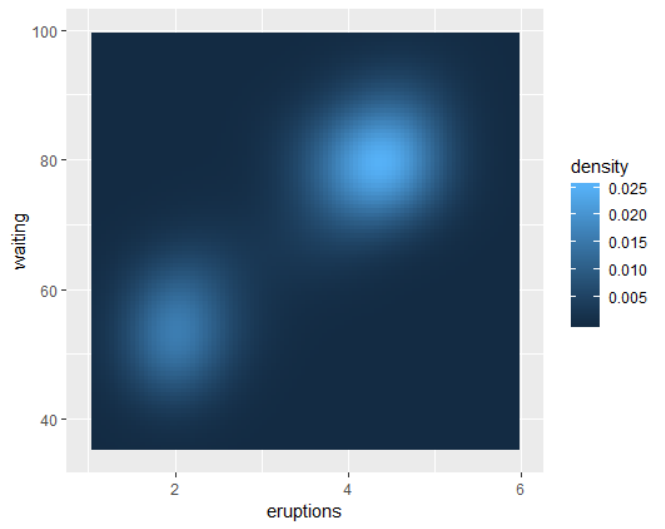


- 높이가 같은 영역을 구분된 색으로 채우는 그래프

```
> p3 + stat_density_2d(aes(fill=stat(level)), geom="polygon")
```

```
> p3 + stat_density_2d(aes(fill=stat(density)), geom="raster",
                       contour=FALSE)
```





4) 산점도 행렬

- 여러 변수로 이루어진 자료에서 두 변수끼리 짝을 지어 작성된 산점도를 행렬 형태로 표현한 그래프

- 자료 분석에서 필수적인 그래프

작성 함수

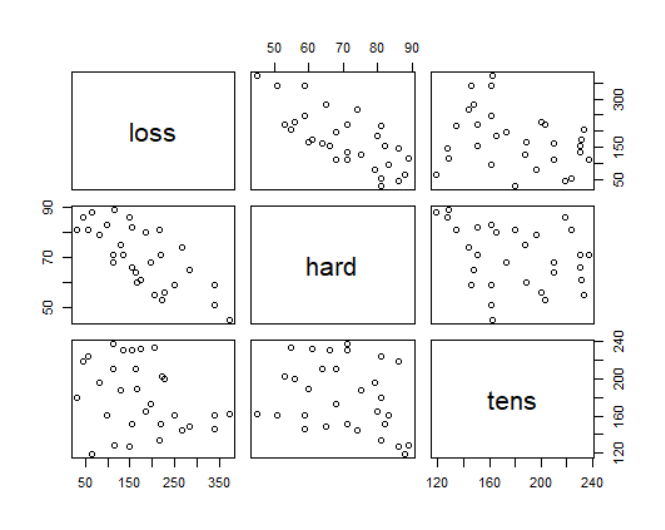
- Base R 그래픽스: 함수 pairs( )

- 패키지 GGally의 함수 ggpairs( )

- 함수 pairs( )에 의한 산점도 행렬 작성

① MASS::Rubber의 세 변수 loss, tens, hard의 산점도 행렬

```
> data(Rubber, package="MASS")
> pairs(Rubber)
```

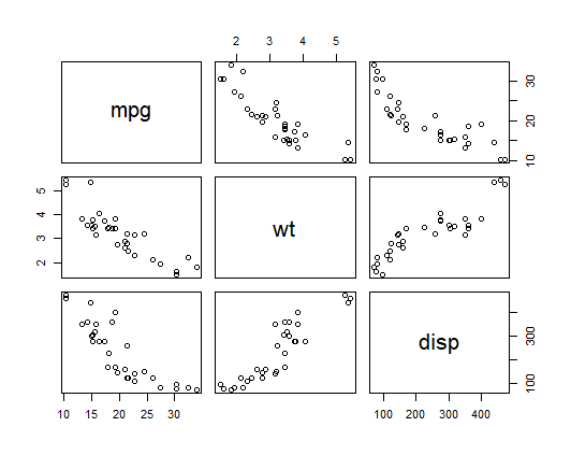


pairs(df): 모든 변수 포함

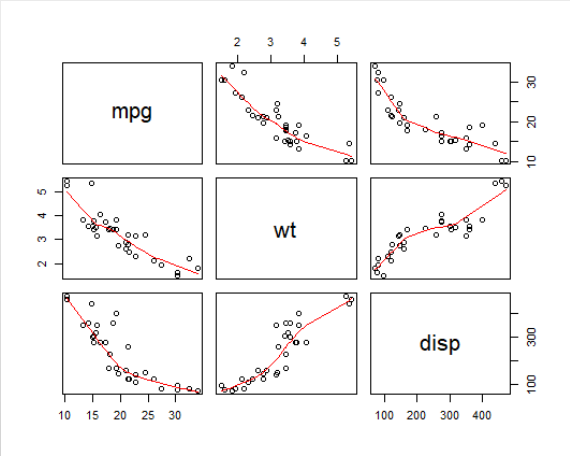
pairs(~ x+ y + z, data=df): 특정 변수 지정

② mtcars의 mpg, wt, disp의 산점도 행렬

```
> pairs(~ mpg + wt + disp, data=mtcars)
```



- 옵션 panel: 패널에 작성되는 그래프의 실질적인 작성
- panel=panel.smooth: 산점도에 국소 선형회귀곡선 추가

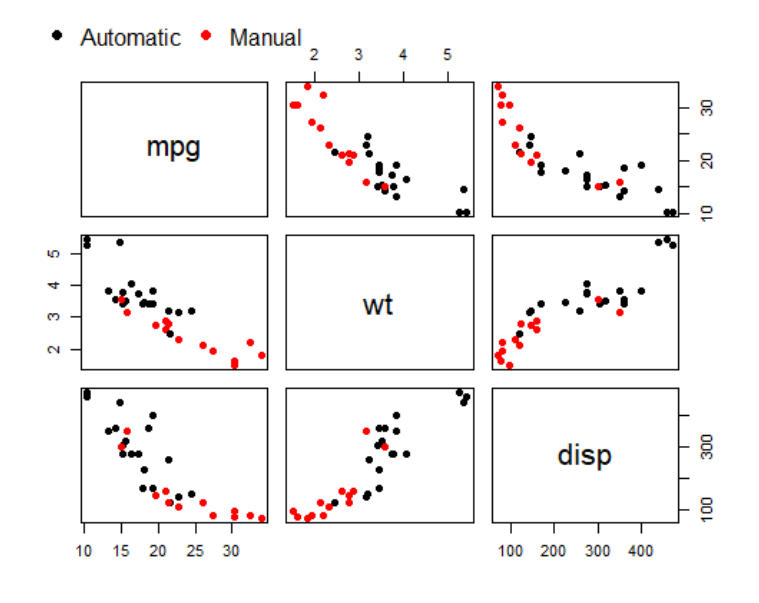



```
> pairs(~ mpg + wt + disp, data=mtcars,
        panel=panel.smooth)
```

③ 사용자가 패널 함수 정의

- mtcars의 변수 mpg, wt, disp의 산점도 행렬
- 변수 am이 0(automatic)이면 검은 점, 1(manual)이면 빨간 점으로 작성

```
> my_panel_1 <- function(x, y) points(x, y, col=mtcars$am+1, pch=16)
> pairs(~ mpg + wt + disp, data=mtcars, panel=my_panel_1)
> legend("topleft", c("Automatic", "Manual"), pch=16, col=c(1,2),
         xpd=TRUE, horiz=TRUE, bty="n", y.intersp=-1)
```

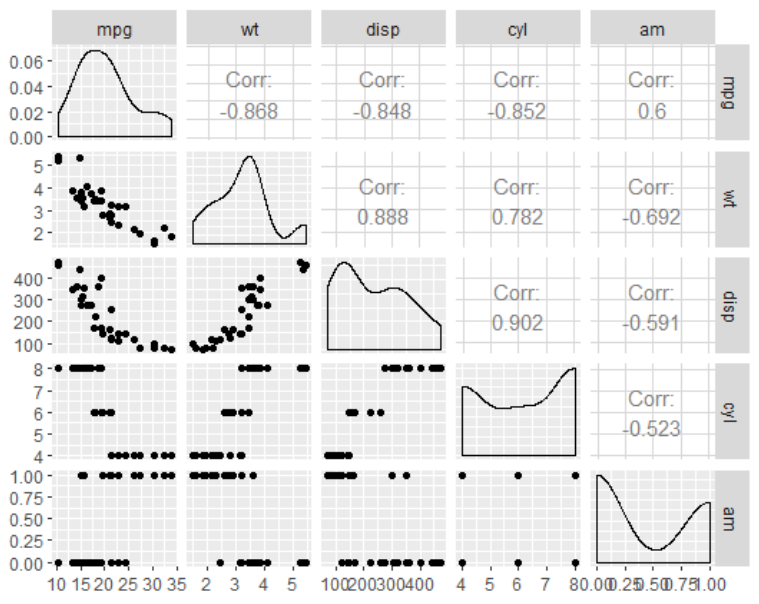


- GGally::ggpairs( )에 의한 산점도 행렬

  - mtcars의 변수 mpg, wt, disp, cyl, am의 산점도 행렬 작성

  - 사용법: ggpairs(df)

```
> mtcars_1 <- mtcars %>%
    select(mpg, wt, disp, cyl, am)
> ggpairs(mtcars_1)
```

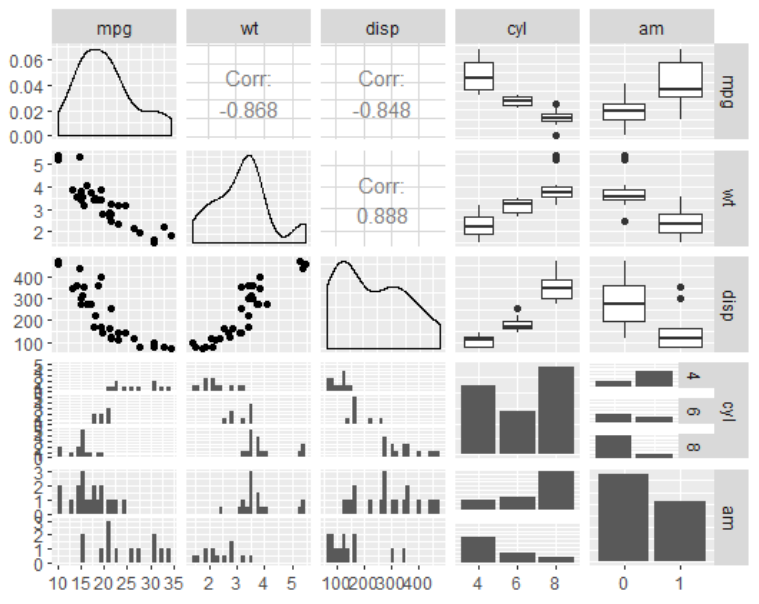


- 모든 변수는 숫자형

- am, cyl를 요인으로 전환

- mtcars의 변수 mpg, wt, disp, cyl, am의 산점도 행렬 작성
- cyl, am을 요인으로 전환

```
> mtcars_2 <- mtcars_1 %>%
    mutate(am=factor(am), cyl=factor(cyl))
> ggpairs(mtcars_2)
```



각 패널에 작성되는 디폴트 그래프

- 대각선 패널: 숫자형(확률밀도 그래프), 범주형(막대그래프)
- 대각선 위쪽 패널: 숫자형(상관계수), 범주형(facet 막대 그래프), combo(상자그림)
- 대각선 아래쪽 패널: 숫자형(산점도), 범주형(facet 막대 그래프), combo(facet 히스토그램)

- 각 패널에 작성되는 디폴트 그래프의 변경

  대각선 위 아래 패널: 옵션 upper, lower

  upper=list(continuous=, combo=, discrete=)

  lower=list(continuous=, combo=, discrete=)

  - continuous: "points", "smooth", "smooth_loess", "density", "cor", "blank"
  - discrete: "facetbar", "ratio", "blank"
  - combo: "box", "dot", "facethist", "facetdensity" , "denstrip", "blank"

  대각선 패널: 옵션 diag

  diag=list(continuous=, discrete=)

  - continuous: "densityDiag", "barDiag", "blankDiag"
  - discrete: "barDiag", "blankDiag"

  그래프 작성의 디폴트 값 변경

  wrap( )

  - 예: wrap("facethist", bins=10)

변경 내용

- 대각선 아래쪽 패널 수정
  숫자형 변수:산점도에 회귀직선 추가
  combo: facet 히스토그램의 구간 개수를 10개로 지정

- 시각적 요소 color에 요인 am 매핑

```
> ggpairs(mtcars_2, aes(color=am),
          lower=list(continuous=wrap("smooth", se=FALSE),
                     combo=wrap("facethist", bins=10)))
```